

APPENDIX

SOLVING MILP

Finding a solution to the MILP expressed by Fig. 3 and Fig. 5 is challenging because (i) the number of variables and constraints is large and (ii) BIG is much larger than the other constants causing numerical issues. Therefore, we will rewrite the MILP to avoid numerical issues. We will also present different methods for solving the MILP; they differ in (i) the amount of time to finish and (ii) whether a solution is guaranteed to be found if a solution exists. They all have in common, however, that they return a tuple $\langle \text{flag}, o \rangle$ such that if flag is true, then the MILP is feasible.

It can be seen in Fig. 5, that changing the domain of $\text{mb}_{i,j,g,b}$ from non-negative integer to non-negative real does not change the feasibility of the MILP. The same applies to $\text{mmb}_{i,i',j',g',b}$, $\text{mmbo}_{i,j,g,b}$, $\text{oat}_{i,j,g,b}$, $\text{oao}_{i,j,g,b}$. We will now rewrite the constraints. Let $\text{SCALINGFACTORNACCESSES}$ be an integer that we choose (e.g. $\text{SCALINGFACTORNACCESSES} = 2^{23}$). Let BUSCLOCKFREQ denote the clock frequency of the bus (e.g. $\text{BUSCLOCKFREQ} = 1.5 \times 10^9$). Then we can introduce $\text{coat}_{i,j,g,b} = \text{coat}_{i,j,g,b} \times \text{BUSCLOCKFREQ}$ and then replace $\text{coat}_{i,j,g,b}$ with $\text{coat}_{i,j,g,b}$. Then we can introduce $\text{mb}'_{i,j,g,b} = \text{mb}_{i,j,g,b} / \text{SCALINGFACTORNACCESSES}$ and then replace $\text{mb}_{i,j,g,b}$ with $\text{mb}'_{i,j,g,b}$. By choosing $\text{SCALINGFACTORNACCESSES}$ properly, we obtain that the variables are in reasonably small range (e.g. six orders of magnitude) and this avoids numerical issues. For convenience, we also rename $\text{coat}_{i,j,g,b}$ and $\text{coat}_{i,j,g,b}$ and rename $\text{mb}'_{i,j,g,b}$ as $\text{mb}_{i,j,g,b}$. This leaves us with discussion on how to choose $\text{SCALINGFACTORNACCESSES}$. We do it as follows.

- 1) $\text{SCALINGFACTORNACCESSES} := 1$
- 2) **if** (69) > 0 **then**
- 3) $\text{SCALINGFACTORNACCESSES} := \text{smallest number} \geq$
- 4) $(70)/(71)$ such that it is equal to two raised to some integer.
- 5) **end if**

We will now present the methods.

Method 1

Method 1 is guaranteed to output a solution if a solution exists. Method 1 is to simply take the constraints in Fig. 3 and Fig. 5 and solve the MILP. If there exists an assignment of values to the variables so that the constraints in Fig. 3 and Fig. 5 are satisfied then flag is true and o is the values of the o -variables; otherwise flag is false and o is undefined.

Method 2

Method 2 is guaranteed to output a solution if a solution exists. We can reason as follows: If there is a feasible solution, then it holds that for each cache color, the pages that are mapped to frames of this cache color all belong to the same task (otherwise (46) would be violated). Let $\text{occupiescachecolor}_{i,h}$ be 1 if task τ_i occupies cache color h ; otherwise 0. If, for this solution, it holds that there is a task τ_i and a task $\tau_{i'}$ and a cache color h and a cache color h' such that $i < i'$ and $h > h'$ and $\text{occupiescachecolor}_{i,h} = 1$ and $\text{occupiescachecolor}_{i',h'} = 1$, then we can change the o -values of the solution so that each page of τ_i that was mapped to h is mapped to h' and each page of $\tau_{i'}$ that was mapped to h' is mapped to h . Also update the x -values accordingly. This gives us a new feasible solution such that $i < i'$ and $h < h'$ and $\text{occupiescachecolor}_{i,h} = 1$ and $\text{occupiescachecolor}_{i',h'} = 1$. Repeating this argument yields that for each task τ_i , tasks with lower index than τ_i only occupies cache colors of lower index and tasks with higher index than τ_i only

occupies cache colors of higher index. If there is a cache color h that is not occupied by any task, then we can identify all tasks that occupies cache colors of index greater than h and let each of their memory allocation use a cache color that has index 1 less. Also update the x -values accordingly.

For this reason, we can, without loss of generality, add the following constraint:

$$\begin{aligned} \forall \langle i', j', g', h', i'', j'', g'', h'' \rangle \text{ s. t. } (\tau_{i'} \in \tau) \wedge (j' \in [1, \text{nstages}_{i'}]) \wedge \\ (g' \in [1, \text{nseg}_{i',j'}]) \wedge (h' \in [0, H-1]) \wedge (\tau_{i''} \in \tau) \wedge (j'' \in [1, \text{nstages}_{i''}]) \wedge \\ (g'' \in [1, \text{nseg}_{i'',j''}]) \wedge (h'' \in [0, H-1]) \wedge \\ (i' < i'') \wedge (h' \geq h'') : x_{i',j',g',h'} + x_{i'',j'',g'',h''} \leq 1 \end{aligned}$$

Method 2 is like Method 1 but with the constraint above.

Method 3

Method 3 is not guaranteed to output a solution if a solution exists. Method 3 is defined as follows.

- 1) Let the following variables be non-negative real numbers: loadfactorofcells , $\text{utilconsideringcont}$, loadofdeadline_i , myobj
- 2) Let $\text{utilconsideringcont}$, loadofdeadline_i be defined as follows: $\text{utilconsideringcont} = (\sum_{\tau_{i'} \in \tau} \frac{\text{cu}_{i'}}{T_{i'}}) / (m \times s)$ and $\text{loadofdeadline}_i = (\sum_{\tau_{i'} \in \tau} \max(\lfloor \frac{D_i - D_{i'}}{T_{i'}} \rfloor + 1, 0) \times \text{cu}_{i'}) / (m \times s \times D_i)$.
- 3) Solve the following problem: minimize myobj subject to the constraints in Fig. 5 and $\forall \langle h, b \rangle \text{ s. t. } (h \in [0, H-1]) \wedge (b \in [0, B-1]) :$
 $(\sum_{\tau_i \in \tau} \sum_{j \in [1, \text{nstages}_i]} \sum_{g \in [1, \text{nseg}_{i,j}]} \sum_{p \in [0, \text{np}_{i,j}-1]} (\frac{1}{\text{GS}_{i,j,g,p}} \times o_{i,j,g,p,h,b})) \leq \text{CAP} \times \text{loadfactorofcells}$ and $\text{loadfactorofcells} \leq \text{myobj}$ and $\text{utilconsideringcont} \leq \text{myobj}$ and $\forall \tau_i \in \tau : \text{loadofdeadline}_i \leq \text{myobj}$.
- 4) Consider the optimization problem of $\text{fmem}(\tau, \Pi, K)$ where the o -values must be equal to the values obtained in step 3 above. Solve this optimization problem.
- 5) If the optimization problem in step 4 is feasible then return $\langle \text{true}, o \rangle$ where o is the o -values obtained in step 3 above.
- 6) If the optimization problem in step 4 is infeasible then return $\langle \text{false}, o \rangle$ where o is undefined.

When solving the first optimization problem, if an optimal solution has not been bound after 3600 seconds, then we terminate and deliver the best result (non-optimal result) so far.

EVALUATION

In this section, we address the following questions: (i) how long time does it take to perform the schedulability test (solve the MILP), (ii) how pessimistic is our schedulability test and (iii) how does the guarantee of our schedulability test compare to the actual behavior in practice (the execution of a program on a real computer).

Consider the system in Fig. 6. It models a hypothetical autonomous system with 4 processors and task τ_1 performing sensor fusion (it first reads the sensors in its 1st stage and then performs parallel processing in its 2nd stage and then merges the results in its 3rd stage) and task τ_2 is a mission controller task (it takes high-level decisions about the mission, e.g. whether the mission should be aborted) and task τ_3 recomputes the current plans when a certain critical event occurs (its 2nd stage performs computations in parallel).

Table II shows the outcome of our evaluation. Each row shows one system and its corresponding outcome. The first column shows the value of $C_{1,2}$. The second column shows the number of memory accesses to a page relative to the number of memory accesses stated in Fig. 6. If the value in the column is 1 then the number of memory

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 18 OCT 2014		2. REPORT TYPE N/A		3. DATES COVERED	
4. TITLE AND SUBTITLE Supplementary material for the paper "Scheduling Constrained-Deadline Sporadic Parallel"				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Andersson /Bjorn A.				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited.					
13. SUPPLEMENTARY NOTES The original document contains color images.					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 3	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

$C_{1,2}$ (seconds)	multi $MA_{i,j,p}$	Time (seconds)	Schedulable
0.005000	0.000000	358.880339	1
0.005000	0.010000	1249.950361	1
0.005000	0.100000	1256.071147	1
0.005000	0.250000	1257.526272	1
0.005000	0.500000	1285.062305	1
0.005000	1.000000	1254.594339	1
0.005000	2.000000	1281.826845	1
0.005000	4.000000	1226.284894	0
0.005000	10.000000	1226.322489	0
0.010000	0.000000	357.445275	1
0.010000	0.010000	1254.029591	1
0.010000	0.100000	1291.771152	1
0.010000	0.250000	1255.745916	1
0.010000	0.500000	1250.537034	1
0.010000	1.000000	1257.315165	1
0.010000	2.000000	1226.780883	0
0.010000	4.000000	1226.388839	0
0.010000	10.000000	1226.468599	0
0.015000	0.000000	357.990881	1
0.015000	0.010000	1250.979049	1
0.015000	0.100000	1250.253606	1
0.015000	0.250000	1264.358976	1
0.015000	0.500000	1255.129114	1
0.015000	1.000000	1255.120019	1
0.015000	2.000000	1226.260251	0
0.015000	4.000000	1226.514951	0
0.015000	10.000000	1226.588321	0
0.020000	0.000000	358.108685	1
0.020000	0.010000	1251.220622	1
0.020000	0.100000	1257.066002	1
0.020000	0.250000	1304.179088	1
0.020000	0.500000	1256.614470	1
0.020000	1.000000	1288.812738	1
0.020000	2.000000	1226.178101	0
0.020000	4.000000	1226.476176	0
0.020000	10.000000	1226.665985	0
0.025000	0.000000	357.629733	1

TABLE II: Results from evaluation ($m = 4$).

accesses to a page is equal to the number of memory accesses stated in Fig. 6. The third column indicates the amount of time it takes to perform the schedulability analysis (with the MILP). The fourth column indicates whether that schedulability analysis provides a guarantee that the taskset is schedulable.

ACKNOWLEDGMENT

Copyright 2014 Carnegie Mellon University This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense. NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT. This material has been approved for public release and unlimited distribution. DM-0001770

$m = 4$	$s=1$	$\tau = \{\tau_1, \tau_2, \tau_3\}$			
$T_1=0.100$	$D_1=0.100$	$nstages_1=3$	$nseg_{1,1}=1$ $C_{1,1} = 0.001$ $np_{1,1} = 17$ $MA_{1,1,0} = 100$ $MA_{1,1,1} = 100$ $MA_{1,1,2} = 100$ $MA_{1,1,3} = 100$ $MA_{1,1,4} = 100$ $MA_{1,1,5} = 100$ $MA_{1,1,6} = 100$ $MA_{1,1,7} = 100$ $MA_{1,1,8} = 100$ $MA_{1,1,9} = 100$ $MA_{1,1,10} = 100$ $MA_{1,1,11} = 100$ $MA_{1,1,12} = 100$ $MA_{1,1,13} = 100$ $MA_{1,1,14} = 100$ $MA_{1,1,15} = 100$ $MA_{1,1,16} = 100$ $MA_{1,1,17} = 100$	$nseg_{1,2} = 4$ $C_{1,2} = 0.030$ $np_{1,2} = 17$ $MA_{1,2,0} = 1000$ $MA_{1,2,1} = 1000$ $MA_{1,2,2} = 1000$ $MA_{1,2,3} = 1000$ $MA_{1,2,4} = 1000$ $MA_{1,2,5} = 1000$ $MA_{1,2,6} = 1000$ $MA_{1,2,7} = 1000$ $MA_{1,2,8} = 1000$ $MA_{1,2,9} = 1000$ $MA_{1,2,10} = 1000$ $MA_{1,2,11} = 1000$ $MA_{1,2,12} = 1000$ $MA_{1,2,13} = 1000$ $MA_{1,2,14} = 1000$ $MA_{1,2,15} = 1000$ $MA_{1,2,16} = 1000$ $MA_{1,2,17} = 1000$	$nseg_{1,3}=1$ $C_{1,3} = 0.001$ $np_{1,3} = 17$ $MA_{1,3,0} = 100$ $MA_{1,3,1} = 100$ $MA_{1,3,2} = 100$ $MA_{1,3,3} = 100$ $MA_{1,3,4} = 100$ $MA_{1,3,5} = 100$ $MA_{1,3,6} = 100$ $MA_{1,3,7} = 100$ $MA_{1,3,8} = 100$ $MA_{1,3,9} = 100$ $MA_{1,3,10} = 100$ $MA_{1,3,11} = 100$ $MA_{1,3,12} = 100$ $MA_{1,3,13} = 100$ $MA_{1,3,14} = 100$ $MA_{1,3,15} = 100$ $MA_{1,3,16} = 100$ $MA_{1,3,17} = 100$
$T_2=0.010$	$D_2=0.010$	$nstages_2=1$	$nseg_{2,1}=1$ $C_{2,1} = 0.002$ $np_{2,1} = 17$ $MA_{2,1,0} = 100$ $MA_{2,1,1} = 100$ $MA_{2,1,2} = 100$ $MA_{2,1,3} = 100$ $MA_{2,1,4} = 100$ $MA_{2,1,5} = 100$ $MA_{2,1,6} = 100$ $MA_{2,1,7} = 100$ $MA_{2,1,8} = 100$ $MA_{2,1,9} = 100$ $MA_{2,1,10} = 100$ $MA_{2,1,11} = 100$ $MA_{2,1,12} = 100$ $MA_{2,1,13} = 100$ $MA_{2,1,14} = 100$ $MA_{2,1,15} = 100$ $MA_{2,1,16} = 100$ $MA_{2,1,17} = 100$		
$T_3=5.000$	$D_3=0.021$	$nstages_3=3$	$nseg_{3,1}=1$ $C_{3,1} = 0.002$ $np_{3,1} = 17$ $MA_{3,1,0} = 100$ $MA_{3,1,1} = 100$ $MA_{3,1,2} = 100$ $MA_{3,1,3} = 100$ $MA_{3,1,4} = 100$ $MA_{3,1,5} = 100$ $MA_{3,1,6} = 100$ $MA_{3,1,7} = 100$ $MA_{3,1,8} = 100$ $MA_{3,1,9} = 100$ $MA_{3,1,10} = 100$ $MA_{3,1,11} = 100$ $MA_{3,1,12} = 100$ $MA_{3,1,13} = 100$ $MA_{3,1,14} = 100$ $MA_{3,1,15} = 100$ $MA_{3,1,16} = 100$ $MA_{3,1,17} = 100$	$nseg_{3,2}=2$ $C_{3,2} = 0.006$ $np_{3,2} = 17$ $MA_{3,2,0} = 100$ $MA_{3,2,1} = 100$ $MA_{3,2,2} = 100$ $MA_{3,2,3} = 100$ $MA_{3,2,4} = 100$ $MA_{3,2,5} = 100$ $MA_{3,2,6} = 100$ $MA_{3,2,7} = 100$ $MA_{3,2,8} = 100$ $MA_{3,2,9} = 100$ $MA_{3,2,10} = 100$ $MA_{3,2,11} = 100$ $MA_{3,2,12} = 100$ $MA_{3,2,13} = 100$ $MA_{3,2,14} = 100$ $MA_{3,2,15} = 100$ $MA_{3,2,16} = 100$ $MA_{3,2,17} = 100$	$nseg_{3,3}=1$ $C_{3,3} = 0.002$ $np_{3,3} = 17$ $MA_{3,3,0} = 100$ $MA_{3,3,1} = 100$ $MA_{3,3,2} = 100$ $MA_{3,3,3} = 100$ $MA_{3,3,4} = 100$ $MA_{3,3,5} = 100$ $MA_{3,3,6} = 100$ $MA_{3,3,7} = 100$ $MA_{3,3,8} = 100$ $MA_{3,3,9} = 100$ $MA_{3,3,10} = 100$ $MA_{3,3,11} = 100$ $MA_{3,3,12} = 100$ $MA_{3,3,13} = 100$ $MA_{3,3,14} = 100$ $MA_{3,3,15} = 100$ $MA_{3,3,16} = 100$ $MA_{3,3,17} = 100$
$K = 20$	$MEMCAP = 2^{21}$	$H = 32$	$B = 16$	$HWSHARE = 1/4$	$CAP = 2^{10}$
$INO = 970$ $shfr = \{\langle 1, 1, 1, 0, 1, 2, 1, 0 \rangle, \langle 1, 1, 1, 0, 1, 2, 1, 0 \rangle, \langle 1, 1, 1, 2, 1, 2, 3, 0 \rangle, \langle 1, 1, 1, 3, 1, 2, 4, 0 \rangle, \langle 1, 3, 1, 0, 1, 2, 1, 1 \rangle, \langle 1, 3, 1, 1, 1, 2, 2, 1 \rangle, \langle 1, 3, 1, 3, 1, 2, 4, 1 \rangle, \langle 3, 1, 1, 0, 3, 2, 1, 0 \rangle, \langle 3, 1, 1, 1, 3, 2, 2, 0 \rangle, \langle 3, 3, 1, 0, 3, 2, 1, 1 \rangle, \langle 3, 3, 1, 1, 3, 2, 2, 1 \rangle\}$ $tck = 1/(1.5 \times 10^9), trrd = 4, tfaw = 20, wl = 7, bl = 8, twtr = 5, cl = 9, trp = 9, trcd = 9, twr = 10$					

Fig. 6: One of the systems used in our evaluation.